Hybrid Procedural Level Generation Using Wave Function Collapse and Genetic Algorithms

Anton Karpetskyi, Piotr Napieralski $^{[0000-0003-1427-7791]}$, and Dominik Szajerman $^{[0000-0002-4316-5310]}$

Institute of Information Technology, Lodz University of Technology, al. Politechniki 8, 93–590 Lodz, Poland akarpetskiy820gmail.com, piotr.napieralski0p.lodz.pl, dominik.szajerman0p.lodz.pl

Abstract. This paper presents a hybrid procedural content generation (PCG) framework that integrates Wave Function Collapse (WFC) for constraint-based structure synthesis with Genetic Algorithms (GA) for adaptive gameplay optimization. Experimental results in Unity show that the hybrid model improves level playability, structural consistency, and generation efficiency, achieving 56% faster convergence and 73% fewer unplayable levels compared to standalone methods. The approach demonstrates the potential of hybrid PCG techniques to enhance scalability, diversity, and player experience in game content generation.

Keywords: Wave Function Collapse- Genetic Algorithm- Procedural Content Generation- Game Levels- Optimization.

1 Introduction

The rapid evolution of video game development has driven an increasing reliance on procedural content generation (PCG) to meet the growing demand for rich, varied, and player-adaptive game environments [1]. PCG techniques enable automatic, on-the-fly content creation that enhances replayability and reduces production costs. Many modern games, particularly in the roguelike and openworld genres, leverage PCG to ensure long-term player engagement [2]. However, automating level generation presents significant computational and design challenges: generated content must be structurally coherent, balanced, and engaging [3].

Research shows that visual diversity can be systematically analyzed using impression curves to evaluate level aesthetics [4]. Thus, integrating both constraintbased structure and adaptive optimization is crucial for effective PCG [5].

Traditional PCG methods fall into two categories: deterministic constraintbased methods and stochastic heuristic-based approaches. Deterministic techniques like Wave Function Collapse (WFC) ensure local logical consistency but offer limited flexibility for gameplay optimization [6]. In contrast, Genetic Algorithms (GA) treat level generation as an optimization problem, allowing the

adaptation of content to desired gameplay properties [7,8]. However, GAs lack inherent structural constraints, often producing invalid or unplayable designs [9].

Given these complementary strengths and weaknesses, integrating WFC and GA offers a promising hybrid approach [2, 10]. Recent models like *Genetic-WFC* demonstrate that such integration can produce more *playable*, *structured*, and *engaging* environments [2].

This paper proposes a hybrid framework that uses WFC for local constraint satisfaction and GA for global gameplay optimization. Our main contributions are:

- A novel two-stage integration of WFC and GA for level generation.
- A full 3D implementation in the Unity engine.
- Experimental comparison showing improved convergence speed, structural validity, and fitness diversity versus standalone methods.

This work expands on prior models such as Genetic-WFC by reducing computational overhead and emphasizing gameplay-driven design control.

2 Hybrid Approach: Integrating WFC and GA

The hybrid generation process consists of three main stages: (1) initial structure generation using WFC, (2) gameplay-aware optimization using GA, and (3) constraint-preserving repair and validation. WFC first produces a level satisfying local adjacency constraints. This level is then encoded into a chromosome for GA optimization, where selection, crossover, and mutation refine gameplay metrics such as difficulty progression, exploration, and pacing. A repair step ensures that GA-generated variants maintain WFC-enforced rules, producing structurally consistent and playable layouts.

The implementation is realized in Unity, using GeneticSharp for evolutionary operations and Blender for modular asset creation. Navigation is ensured by A* pathfinding based on Unity's NavMesh.

2.1 Initial Constraint Satisfaction (WFC Preprocessing)

The first stage generates a structurally valid initial layout using Wave Function Collapse. The game level is represented as a grid of tiles (e.g., Floor, Wall, Door, Entrance, Exit, Obstacle, Treasure), each with defined directional adjacency rules encoded in a compatibility matrix.

The WFC algorithm collapses tiles by entropy minimization, propagating constraints. It supports 3D layouts with vertical adjacency constraints (e.g., stairs connecting different layers). Fixed boundary conditions and tile frequency weights guide the generation process.

This preprocessing ensures that all candidate levels respect basic structural grammar, allowing the GA phase to focus solely on gameplay optimization.



Fig. 1. Set of tile types used in WFC. Each tile enforces adjacency constraints during level generation.

2.2 Evolutionary Optimization (GA Enhancement)

After WFC generation, levels are optimized using a Genetic Algorithm. The 2D grid is linearized into a 1D chromosome where each gene encodes a tile type.

Population size 50			
Generations	100		
Selection	Tournament (size 3)		
Crossover	Two-point (80% probability)		
Mutation	Random tile mutation $(1\% \text{ per gene})$		
Elitism	Top 2 preserved		

 Table 1. Genetic Algorithm Parameters

The fitness function F(L) evaluates each level L as:

$$F(L) = w_d f_{\text{diff}}(L) + w_e f_{\text{expl}}(L) + w_p f_{\text{prog}}(L),$$

where:

- f_{diff} : difficulty progression,
- f_{expl} : exploration potential,
- f_{prog} : logical progression structure.

Unplayable levels (e.g., disconnected paths) are penalized or discarded. The GA drives optimization toward structurally sound and gameplay-relevant designs.

2.3 Final Level Evaluation

After GA optimization, the final levels undergo a comprehensive validation process. First, the navigability of the environment is verified using A* pathfinding algorithms to ensure that all key objectives and regions are reachable. Next, a structural integrity check confirms that the generated layout complies with adjacency and connectivity constraints. Finally, the gameplay metrics are evaluated to assess progression difficulty, exploration potential, and logical flow throughout the level.



Fig. 2. Comparison of generated levels: (left) GA-only, (middle) WFC-only, (right) Hybrid WFC-GA. The hybrid method balances structure and gameplay.

Parallel fitness evaluations, implemented through Unity's Job System and optimized with the Burst Compiler, significantly reduce computation times during the evolutionary optimization phase. By distributing fitness evaluations across multiple threads and leveraging low-level code optimizations, the hybrid WFC-GA framework achieves near-linear scaling with respect to available CPU cores. This substantial improvement in computational efficiency not only accelerates the convergence process but also enables the practical deployment of the hybrid generation pipeline in real-time or near-real-time game development environments. Consequently, the proposed approach meets the stringent performance requirements of modern interactive applications while maintaining high standards of structural coherence and gameplay quality.

3 Experimental Evaluation

We evaluated the hybrid WFC-GA model in comparison with standalone WFC and GA approaches. The evaluation focused on generation efficiency, structural validity, gameplay quality, and level diversity. All methods were implemented in Unity. The GeneticSharp library handled GA operations, while Unity's Job System and Burst Compiler were used to parallelize fitness evaluations.

Three procedural generation methods were tested: (1) WFC-only, employing constraint-based generation without optimization; (2) GA-only, optimizing gameplay metrics but without enforcing structural rules; and (3) the hybrid WFC-GA approach, integrating constraint satisfaction with evolutionary refinement. Each method generated 30 independent levels on 30×30 grids, using a GA population size of 50 and running for up to 100 generations. The target novelty

score for level variation was set to 0.8. Performance was assessed based on convergence time to fitness plateau, average generation time, structural consistency (percentage of playable levels), and fitness diversity measured via entropy.

The results showed that the hybrid method converged to high-quality solutions significantly faster than GA-only, achieving a 56% improvement in convergence speed. WFC-only generation, while structurally reliable, incurred longer times due to repeated backtracking in highly constrained configurations.



Fig. 3. Average fitness over 100 generations. The hybrid method converges faster and achieves higher fitness.

Table 2 summarizes the average generation time and percentage of invalid (unplayable) levels for each method. The hybrid model notably achieved a lower generation time (8.45 seconds on average) while maintaining high structural validity, producing only 10.1% invalid levels compared to 37.5% in the GA-only approach.

Table 2. Average generation time and invalid level rate.

Method	Time (s)	Invalid Levels (%)
WFC-only	19.40	0.0
GA-only	13.20	37.5
Hybrid WFC-GA	8.45	10.1

Beyond efficiency, the hybrid model demonstrated a clear advantage in structural validity and gameplay quality. While WFC-only guaranteed fully playable structures, it lacked adaptability to gameplay design objectives. Conversely, GA-only produced more gameplay-driven layouts but often failed to maintain traversability. The hybrid method combined the strengths of both, achieving 89.9% fully playable levels, significantly higher than GA-only (62.5%).

Further analysis revealed that hybrid-generated levels exhibited better difficulty progression and exploration structure. Entropy-based measurements of fitness diversity showed that the hybrid approach produced a broader variety of high-quality levels compared to both standalone methods.



Fig. 4. Distribution of final fitness scores. The hybrid method produces more diverse, higher-quality results.

The hybrid WFC-GA model outperformed both baselines by achieving faster convergence, higher structural validity, improved gameplay-driven layouts, and greater diversity in generated levels. These results demonstrate that combining constraint propagation with evolutionary optimization provides a powerful framework for scalable and dynamic procedural content generation.

4 Discussion and Future Work

The results demonstrate that combining constraint-based structure generation (WFC) with evolutionary gameplay optimization (GA) offers a scalable and effective solution for procedural level generation. The hybrid method successfully balances structural validation and gameplay adaptability while maintaining acceptable generation times, making it well-suited for dynamic and complex game development environments.

Standalone WFC requires significantly more computation time due to constraint propagation, while GA, although faster, often sacrifices structural guarantees. As shown in Table 3, the hybrid model achieves an optimal balance between these approaches, enabling efficient and structurally valid level generation without compromising gameplay-driven design objectives.

Table 3. Execution time comparison of procedural generation methods.

Metric	WFC	\mathbf{GA}	Hybrid WFC-GA
Evolution Time (s)	19.40	3.42	8.45
Generation Time (s)	0.51	0.00	0.54

Despite these advances, the current hybrid framework employs static, predefined fitness functions, limiting its ability to adapt dynamically to evolving player behaviors. Future research should incorporate *reinforcement learning* (RL) mechanisms capable of real-time difficulty scaling, personalized content adjustment, and autonomous adaptation of gameplay progression based on player interaction patterns. Integrating such learning-based methods would allow the system to respond flexibly to diverse player strategies, enhancing both engagement and replayability.

Moreover, although the present work focuses on tile-based environments, the hybrid WFC-GA framework holds potential for extension to a wider range of game genres. Promising directions include open-world terrain generation through procedural heightmaps, the synthesis of three-dimensional dungeon layouts with vertical connectivity, and the development of narrative-driven procedural worlds where AI-generated story arcs influence environment design. Realizing these extensions will require expanding constraint models, redesigning fitness objectives, and integrating additional semantic layers into the generation process.

In conclusion, the hybrid WFC-GA approach provides a robust and extensible foundation for the next generation of procedural content generation systems. Future work will focus on enhancing computational scalability, introducing player-adaptive real-time generation capabilities, and broadening applicability to complex, narrative-rich virtual environments.

5 Conclusion

This paper presented a hybrid procedural content generation approach that integrates *Wave Function Collapse (WFC)* for constraint-based structure generation with *Genetic Algorithms (GA)* for adaptive gameplay optimization. The proposed method successfully balances local structural coherence with global design objectives, addressing key limitations of traditional standalone techniques.

Experimental evaluation conducted in Unity demonstrated that the hybrid model significantly reduces generation time compared to standalone WFC, ensures the production of structurally valid and playable levels, and achieves greater

gameplay diversity and player engagement. These findings underscore the effectiveness of combining deterministic constraint propagation with stochastic evolutionary search in procedural generation pipelines.

Future research directions include integrating reinforcement learning to enable real-time adaptive content generation based on player behavior, further improving scalability through parallelization strategies, and extending the framework to support open-world, three-dimensional, and narrative-driven game environments. Overall, the hybrid WFC-GA model establishes a robust foundation for the next generation of scalable, dynamic, and player-responsive procedural content generation systems.

References

- Shaker, N., Togelius, J., Nelson, M.J.: Procedural Content Generation in Games: A Textbook and an Overview of Current Research. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-42716-4
- Bailly, R., Levieux, G.: Genetic-WFC: Extending Wave Function Collapse With Genetic Search. *IEEE Transactions on Games*, vol. 15, no. 1, pp. 36–45, March 2023. https://doi.org/10.1109/TG.2022.3192930
- Moghadam, A.B., Rafsanjani, M.K.: A Genetic Approach in Procedural Content Generation for Platformer Games Level Creation. In: Proc. 2nd Conf. Swarm Intelligence and Evolutionary Computation (CSIEC), pp. 141–146 (2017). https://doi.org/10.1109/CSIEC.2017.7940160
- Andrzejczak, J., Osowicz, M., Szrajber, R.: Impression Curve as a New Tool in the Study of Visual Diversity of Computer Game Levels. In: Krzhizhanovskaya, V.V. et al. (eds.) Computational Science – ICCS 2020, LNCS, vol. 12141, pp. 499–512. Springer (2020). https://doi.org/10.1007/978-3-030-50426-7\ 39
- Spierewka, L., Szrajber, R., Szajerman, D.: Procedural Level Generation with Difficulty Estimation for Puzzle Games. In: Paszynski, M. et al. (eds.) Computational Science – ICCS 2021, LNCS, vol. 12746, pp. 87–100. Springer (2021). https://doi.org/10.1007/978-3-030-77977-1\ 9
- Kim, H., Lee, S., Lee, H., Hahn, T., Kang, S.: Automatic Generation of Game Content using Graph-based WFC Algorithm. In: Proc. 2019 IEEE Conf. Games (CoG), pp. 1–4 (2019). https://doi.org/10.1109/CIG.2019.8848019
- Liapis, A., Yannakakis, G.N., Togelius, J.: Sentient Sketchbook: Computer-aided Game Level Authoring. In: Proc. Foundations of Digital Games (FDG), pp. 213– 220 (2013).
- Karth, I., Smith, A.M.: WaveFunctionCollapse: Content Generation via Constraint Solving and Machine Learning. *IEEE Transactions on Games*, vol. 14, no. 3, pp. 364–376 (2022). https://doi.org/10.1109/TG.2021.3076368
- 9. Wu, Z., Mao, Y., Li, Q.: Procedural Game Map Generation using Multileveled Cellular Automata. In: Proc. ISAIMS'21, pp. 168–172 (2021). https://doi.org/10.1145/3500931.3500962
- 10. Susanto, E.K., Fachruddin, R., Diputra, M.I., Herumurti, D., Yunanto, A.A.: Maze Generation Based on Difficulty using Genetic Algorithm with Gene Pool. In: Proc. iSemantic 2020, pp. 554–559 (2020). https://doi.org/10.1109/iSemantic50169.2020.9234216